**CS6456 (F2015) Operating Systems**

**Project 3**

**Title: Sudoku Solution Validator**

**Due: September 30, 2015 (11:00 a.m.)**

**Points: 10**

Write a program in C/C++ to determine whether the solution to a Sudoku puzzle is valid (p. 197). Your program should be designed in such a way that it contains multiple threads working together to determine the answer.

(1) The project is not a team work.

(2) Your program will be compiled and run in the Debian VM environment.

(3) You should read the problem shown on pp. 197-199 carefully and understand thoroughly what the problem your program will solve.

(4) A valid Sudoku puzzle is one that each row and each column contains all of the digits 1..9 and each 3×3 subgrid (a total of nine) also contains all of the digits from 1..9.

(5) You should use multithreaded approach shown in class to design your program. You may follow the strategy suggested in the book or use a design methodology different from the one shown in the book. Regardless of what design methodology you are using, your program should contain at least the following threads: one thread to check that each row contains all of the digits 1..9, one thread to check that each column contains all of the digits 1..9, and nine threads to check that each of the nine 3×3 subgrids contains all of the digits 1..9.

(6) The digits in the 9×9 grid are stored in a file named **sudoku.txt**. The digits are stored in the file in a contiguous way, with each pair of digits separated by exactly one space. Your program will read in all the digits from the file and store them in a two-dimensional array with 9 rows and 9 columns in a row-major manner. For example, the first digit is stored in row 1 and column 1, the second digit is stored in row 1 and column 2, the third digit is stored in row 1 and column 3, and so on. You may assume that all of the digits stored in the file are of **int** type with values in the range of 1..9 (inclusive).

(7) Your program should be able to cover all of the following cases. (i) If a digit is missing in a 3×3 subgrid, print a message indicating the position of the 3×3 subgrid (e.g. rows 1..3 and columns 4..6) and the digit that is missing in the subgrid. If multiple digits are missing in the subgrid, your program should print the position of the 3×3 subgrid, followed by all the missing digits. (ii) If a digit in a row or in a column is missing in the 9×9 grid, print the row number or the column number corresponding to the missing digit. Your program should be able to identify and print all the digits that are missing in the row or the column. (iii) If each row and each column in the 9×9 grid contains all of the digits 1..9, print a message indicating that it is a valid Sudoku.

(8) You should use a make file to compile your program.

(9) Turn your project in via **collab** in a **tar** file (created using the **tar** command) consisting of (i) all headers accessed in the source code file; (ii) source code file for project 3; (iii) a make file; and (4) a write-up showing your design and implementation of the project. The **tar** file should be named as follows: **p3**, followed by the first letter of your first name, followed by your last name, and followed by the file extension (i.e. **tar**). For example, the tar file turned in by **John Smith** should be named **p3jsmith.tar**.

(10) Submit also a hardcopy of the **tar** file in class consisting of the four items shown above.

(11) Your program should be successfully compiled. Programs failing to compile will not get any points.